INTERNATIONAL
HELLENIC
UNIVERSITY

Financial Nonlinear Time-Series Analysis and Prediction with Reservoir Computing

Panagiotis Tziatzios
SID: 3308180026

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Data Science

DECEMBER 2019

THESSALONIKI – GREECE

Financial Nonlinear Time-Series Analysis and Prediction with Reservoir Computing

Panagiotis Tziatzios

SID: 3308180026

Supervisor: Dr. Stavros Stavrinides

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Data Science

DECEMBER 2019

THESSALONIKI – GREECE

Abstract

This dissertation was written as part of the MSc in Data Science at the International Hellenic University. The scope of this dissertation is to introduce the reader to Time-series analysis using Reservoir Computing after establishing the core concepts of Time-Series, Chaos theory and Reservoir Computing. Furthermore, a brief introduction of Neural Networks is going to be discussed. More specifically, a more special tool of Reservoir Computing, Echo State Networks, is going to be thoroughly explained and used for the prediction of real, raw financial data. For that, sample data of both financial stocks and indices will be used to validate the potential of the model. Finally, the accuracy of the Echo State Networks proposed are compared to the random heuristic approach.

Panagiotis Tziatzios

02/12/2019

Acknowledgements

At this point I would like to personally thank my supervisor Dr. Stavros Stavrinides for his valuable help and instructions throughout this work. His proposal for the concept of nonlinear dynamics and reservoir computing was a great prospect and helped me explore this wonderful field of science. I would also like to thank Mr. Ioannis Antoniadis for his co-work with me and his helpful advice for every piece of my master thesis. Finally, I must thank my MSc in Data Science fellow and good friend, Spyridon Georgopoulos for working with me from the start of this thesis and for his innovative ideas and patience.

Table of Contents

1 Introduction

1.1 General introduction - Scope of the dissertation

From the beginning of the world people always have wanted to develop techniques that would allow them produce tools for predicting the future. Weather forecasting for example is such a case and hence mariners were urged to create innovative heuristic ideas to forecast the weather. Knowing the future, or being able to make a prediction about it, has helped mariners in this case, to plan their journeys and get better prepared for bad destructive weather conditions. Apart from seamen, people in general have developed successful heuristic ideas that have served their purpose and helped them make better decisions based on these predictions. Nowadays, that the technology has been developed and rapid changes have been made in many scientific fields, there is a bigger thirst for developing new, more effective tools that would approximate the real world more successfully. And since making profit is what people care about the most , the forecasting of financial situations could not be absent from research. Financial data exhibit a strange chaotic behavior and hence the history of this behavior needs to be exploited to gain valuable insight about the trend of the financial market.

The scope of this dissertation is to give the reader the basis of Time-Series and delve into reservoir computing as a tool for forecasting Financial Time-Series. In the beginning, some of the past research is going to be presented to have a knowledge of what has been done as far as reservoir computing and neural networks for financial forecasting is concerned. Then Echo State Network (ESN), a version of RC, will be the subject of study along with Recurrent Neural Networks and nonlinear time series analysis. Finally, the results of the research are going to be presented and discussed. The study was conducted using both noise-free datasets (Vosvrda system) and real financial data like stock price evolution of American Airlines and the index S&P 500. The real raw data were used from *yahoo.finance* and refer to the period of 10 years starting from September 2009. At first noise-free time series are going to be discussed in order to give the reader a first impression of the way an Echo State Network performs and then move to real raw financial data and the approach to predict the daily Close value of a stock.

1.2 Time Series

When a research is conducted the first thing a researcher does is to look for historical data or past research in order to initialize their study. Hence, the evolution of a system is useful to extract meaningful patterns and analyze the behavior of the data through time. Time series is a collection of data points listed in time order, in such a way that they are equivalently spaced in time. Time series analysis consists of methods/tools so as to extract the characteristics of the data whereas time series forecasting uses the analysis in order to build models that estimate the progress of the time series in the future[1]. Time Series can be divided into two subcategories: Univariate-Multivariate. A Univariate time-series refers to the evolution of the values of a single (scalar) variable through time. Again, the time-interval is equal for every timestep t and thus time is not an extra variable. Multivariate time series, as its name suggests, are described by many individual variables that all together describe a bigger quantity like a stock or an index. An example of a multivariate time series can be the growth of a stock price, that is described by many characteristics on a daily basis, e.g. its Open and Close value as well as its High and Low value during the day. A univariate time series on the other hand can be considered to be the daily outdoor temperature, which is measured every day [2] and can be perceived graphically as a line plot as shown in Figure 1-1.

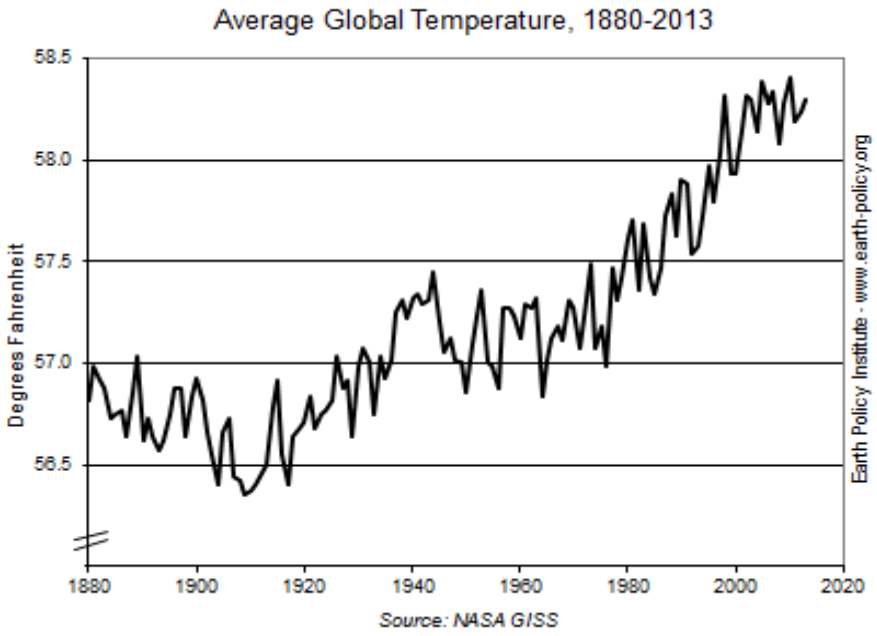

Figure 1-1. Univariate time series[3]

1.3 Neural Networks

Neural Networks has been in the center of research in the past decade and has been the most powerful mechanism in solving many tasks of different domains like medicine, finance or sports. Neural nets can be found in different versions, as they were presented by scientists, each one of them being more specialized in different applications. In general, artificial neural networks try to mimic the human brain as they exhibit similar functionality. In their most simplified version they consist of three layers: the input layer, where the initial data are fed into the network, the hidden layer(s),where the original data are transformed according to the value of an activation function and the output layer, where the output of the network is produced. At this final step, the output is compared to the desired (true) output and the intelligence of the network takes place. It tries to improve its performance by propagating the error back through the connections(synapses) between the nodes(neurons) trying to minimize the error, based on a loss function. The most common version of Neural Networks for manipulating financial data and forecasting is Recurrent Neural Networks. A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence.[4]

1.4 Reservoir Computing

Reservoir computing can be defined as an extension of typical neural networks[4].This approach, which had been previously spotted in computational neuroscience and was later presented in machine learning as the Backpropagation Decorrelation learning rule, is now increasingly often collectively referred to as Reservoir Computing (RC)[5]. In fact, RC takes the input signal u(t) and feeds it in the so-called reservoir in such a way that it is mapped into higher dimensions. Then using a simple learning algorithm like ridge regression, RC tries to learn the output weights between the mapped data and the true output. The whole philosophy of Reservoir Computing and its variations are going to be discussed further down in detail.

1.5 The need of nonlinear time series analysis

Financial systems are complex dynamical systems that change according to many parameters many of which are unknown. But even those parameters that define how a stock price changes, cannot be measured sometimes because of the difficulty of measurement or the nature of the variable. Political changes for example or unexpected destructive weather phenomena can have a huge impact on the global stock market but cannot be measured quantitatively. So, when we attempt to model and understand such dynamical systems, we must face their complexity and lack of data of course and try to search for analogous tools that might help our research. Nonlinear dynamics provide us such capabilities and can be used to infer valuable information from the data. Dynamical systems are governed by state variables that describe the systems properly at every time step. For example, angular position θ and velocity ω are the state variables of the pendulum and can describe its position fully. So, pendulum needs only two state variables to be described properly by an ordinary differential equation

$$\ddot{\theta} = -g * \sin\theta(t), \text{ (1.1)}$$

,which as can be seen, is a nonlinear differential equation due to the *sin* term [6]. Such dynamical systems can be modeled easily using a formula, but this is not the case in many other interesting situations such as financial systems which are part of this category. The idea is to try to model the behavior of these systems by mapping the given data into the higher dimensional space, gaining that way the missing information of the system due to lack of data. Based on this, reservoir computing is going to be used in depth to forecast the 'Close' value of stock and index and mention other tools of nonlinear dynamics and chaos theory.

2 Related Work

In this chapter there will be highlighted past research on reservoir computing for financial forecasting. Since reservoir computing is somewhat new in the scientific world, as it was first introduced at the beginning of the 21st century, there is not an extensive bibliography on this domain. However, a remarkable job has been done by many notable researchers and it is going to be discussed in the next line.

2.1 Other research results

Since reservoir computing has its roots in nonlinear dynamics and machine learning, many scientists have tried to model chaotic time series and set it as the starting point to model more complex, real-data time series. Mackey-Glass is one of the tritest chaotic time series that has been studied a lot and great results have been achieved. Mackey-Glass is a nonlinear time delay differential equation as described by the formula below:

$$\frac{dx}{dt} = \beta * \frac{x_\tau}{1+x_\tau^n} - \gamma * x, \quad (2.1)$$

where β, γ, τ, n are real numbers, and x_τ represents the value of the variable x at time $(t-\tau)$. Depending on the values of the parameters, this equation displays a range of periodic and chaotic dynamics. [7]. The studies have been focused on how to optimize the parameters of the reservoir to gain the best results with respect to some error metrics. Jenny Su in her master thesis "Reservoir Computing in Forecasting Financial Markets" introduces Mackey-Glass time series and tries to tune the hyperparameters of the network so as to achieve the best Minimum Squared Error (MSE) [8]. The purpose of the study was to familiarize herself and the reader with the concept of Reservoir Computing and then run some experiments to see the performance of the network using different parameters at a time. Furthermore, after having introduced the concept of Echo State Network, the researcher makes the attempt to model financial time series data after the application of some data preprocessing techniques. However, the results from this attempt do not show that echo state networks can significantly predict stock data.

Yufan Wang in his master thesis under the title "Applications of Recurrent Neural Networks on Financial Time Series" tries to test the performance of the Echo State Network on denoised

financial times series[9]. His concept is to use historical data in order to predict next day stock prices and returns . The concept of data preprocessing can be also found in his master thesis and the denoising of the raw financial data seems to be a crucial factor in his research. Konrad Stanek in his work performs a comparison between neural networks and echo state networks and uses financial data to propose a model for forecasting. He gives a great explanatory view of financial data introducing key concepts of the field helping the reader understand the plan of attack. He also feeds his proposed model with additional variables so as to improve the quality and quantity of information that the network receives helping that way the algorithm capture better the dynamics of the time-series [10].

3 Neural Networks-Echo State Networks-Nonlinear Dynamics and Chaos Theory

In the lines to follow there is going to be discussed in detail the concepts of neural networks, nonlinear dynamics and chaos theory. Neural networks exhibit a great architecture that resembles with that of the human brain which is discussed below. Chaos is a controversial word that might be misleading many times. Chaos and periodicity are separated by a thin line and small changes in one or more hyperparameters can set the system's behaviour to either one or the other.

3.1 Neural Networks

3.1.1 Introduction

Neural networks have been one of the most discussed topics in the last twenty years as they have been able to solve many challenging tasks of many scientific fields. The whole philosophy of Artificial Neural Networks (ANN) is based on the architecture of the human brain, that consists of billions of neurons, each of which is connected to each other, delivering messages for specific tasks of the brain. The same perspective holds for the Neural Networks as well. The network consists of an input layer where the original signal is fed into an arbitrary number of neurons i. These input nodes(neurons), relate to the rest network with connecting weights w_{ij} between the input and the hidden layers, which can vary and there is not a general rule that determines the number of them as it depends on the examined task. It should be mentioned here that the nature of the data varies from task to task and it is advisable that the network receives specific form of data. So, data preprocessing techniques like data normalization and data cleaning should take place before training the network in order to make it easier for the network to be trained and not lose useful information. For example, when it comes to financial cases there might be data that refer to different currencies fed into the network which can be very misleading for the activation neurons.

The preprocessed information that every input node carries, is passed forward to the rest of the network using a mathematical formula that determines the value that corresponds to the respective node in the hidden layer. Since every node in the hidden layers is connected with all the input nodes, its value is determined if we summarize all the products between the weights and the value of the node, plus a small bias. More precisely, for every neuron j in the hidden layer it holds that its value is given by:

$$y(j) = f(\Sigma w_j * x_i + b), \quad (3.1)$$

for every input node i where f is an activation function and b is a bias added to the sum as shown in the Figure 3-1 below.[11]

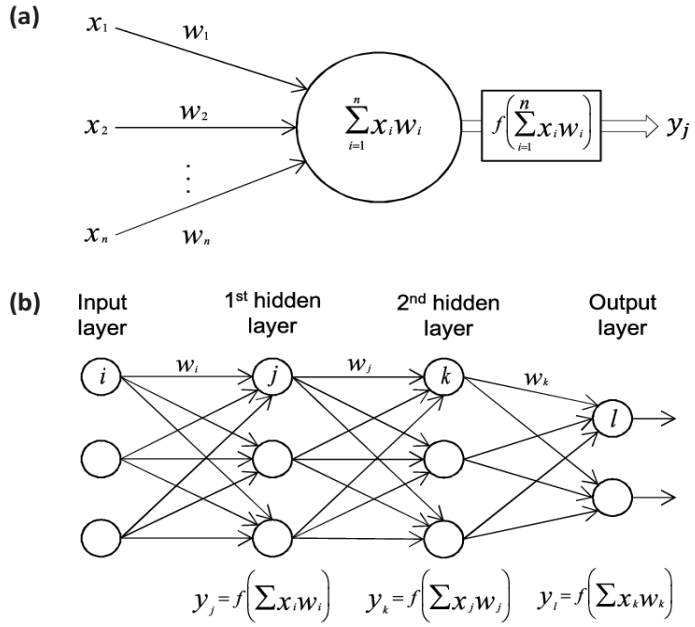

Figure 3-1. *Architecture of a simple feed forward neural network*

3.1.2 Activation functions

As it was mentioned earlier, neural networks try to imitate human brain. When a message in the brain is delivered, different neurons have been previously lit up to pass on the appropriate type of information. Activation functions can be also perceived that way. In fact, these functions define the output of a node based on an input and the respective weights and can determine whether a node will light up or not. Not all neurons are responsible for a specific output and hence not all intermediate neurons are fired up. Apart from that, since the goal is to produce output based on an input, activation functions are used so as to learn to map the input to a desired output. Neural networks under the *universal approximation theorem***,** can approximate any continuous function and so are trained to learn even complex mappings

between input and output. Activation functions play an important role in this, as can preserve the properties of the data if they are carefully chosen. Not all relations, in fact very few, have linear behavior in real world cases. This means, that if we take as input data a quantity *x*, its output *y* will not be equal to $y = k * x + b$ in most of the cases, meaning that it will not behave linearly. So, activation functions should take consideration of such nonlinearities in the data to avoid binary firings of the neurons (ON/OFF) and slow convergence. Furthermore, another property that should be satisfied is that activation functions should be differentiable. This happens mainly because the whole training process of neural networks is based on the backpropagation algorithm, which by using partial derivatives propagates the error back to the input layer. More details on this will be given further down.

3.1.3 Bias

Bias, that was previously spotted in the output equation, can be an additional parameter that adjusts the weighted sum of the input nodes. Bias acts like the constant term *b* in the linear equation $y = a * x + b$ that shifts the line from the origin (0,0) and it is also used in the neural networks to for the training purposes. To be more specific since each node is fired up based on its value and a threshold bias adjusts this threshold accordingly for each neuron. That way it helps the network to find more complex patterns.

3.1.4 Backpropagation

Backpropagation is a training algorithm for supervised learning that uses the concept of gradient descent or stochastic gradient descent. Supervised learning is the method of learning when we try to find a mapping between an input and a true-desired output that is known to us in advance. Backpropagation takes place at the last part of the training process and the idea of the algorithm is based on derivatives. Once the network produces the output, it is then compared to the desired one. Based on a cost function, that measures how much the predicted output differs from the true one, the error is propagated back to check which neurons that were triggered were responsible for the output. The training/learning part comes to the attempt of finding the best weights that minimize the cost function. Such cost functions could be, the Mean Squared Error (MSE), the Mean Absolute Error (MAE) the Root Mean Squared Error (RMSE) and many more depending on the task.

Backpropagation algorithm concludes this small introduction to neural networks that will be used to compare their generalisation ability to reservoir computing. Neural networks have been under research for many decades now and scientists have introduced many types of them. One such type is the Recurrent Neural Networks (RNN) that are used for temporal sequences. This form of neural nets is designed to have and maintain "memory" and therefore are used for tasks where history affects at some point the future. More details about RNN's are not in the scope of this dissertation. Further down, reservoir computing will be presented and discussed in detail, as well as Echo State Networks (ESN) that will prepare the reader for the presentation of my suggestion to forecast real world financial data. In addition to this, since reservoir computing has its roots in nonlinear dynamics, there will also be a brief report on fundamental concepts of nonlinear dynamics and chaos theory.

3.2 Nonlinear Dynamics-Chaos Theory

The idea of this dissertation is based on the concept of dynamical systems and especially the nonlinear ones. A dynamical system is defined to be a system that is governed by an equation, evolving over time. So, there is a time dependence which is represented in the geometrical space and can give at any time the future position of the system. There are mainly two types of dynamical systems, differential equations and iterated maps or difference equations. In the first category of dynamical systems, time is continuous, meaning that there is a non-stop evolution of the system and are widely used in many scientific fields like physics, biology engineering and others. The exponential growth of a population for example, is governed by the equation $\dot{x} = rx,$ where x is the population at time t and r is the growth rate. Population x is in fact a function of time t and hence the dot denotes the derivative of x with respect to t. Difference equations on the other hand refer to discrete time, meaning that time can take any value in the set of natural numbers $\mathbb{N}$. Difference equations or iterated maps as they are also called, can be very useful for analysing chaos as well as simulate and estimate solutions of difficult differential equations. As mentioned before, most of natural phenomena that have concerned scientists for years and can be modelled by a formula, are nonlinear. By "nonlinear" we mean that the unknown variable x occurs in, powers, products, sin, cos etc. For example,

$$\ddot{x} + \frac{g}{L}\sin(x), \quad (3.2)$$

describes the swinging of a pendulum is a nonlinear, second order differential equation, where, g is the gravity, L is the length of the pendulum, x is the angle of the pendulum. Nonlinearity

here is due to the sin term and is what makes it very difficult for the equation to be solved analytically. So, dynamical systems change over time and hence there is a creation of data points out of the mathematical formula that governs the systems. All the possible values that the system can take are defined to be the *phase space* of the system. The mathematical formula that describes fully the possible position of the system at any given time t and so, every variable that appears in the formula is called *state variable.* State variables are the axes in the phase space and if they are known can give the full information of the system's future position. It might sound an easy task to have the knowledge of all the state variables of a system, but it turns out that it is not. In fact, most scientists lack data and try to estimate and approximate the behaviour of a system. Although the technology has been developed rapidly and sensors can give us real-time data, it remains a huge obstacle for people to know all the information needed to describe fully a dynamical system. Such a case can be the harmonic oscillator which is described by the formula: $m\ddot{x} = -kx$, where k is a constant, m the mass of the spring and x is the position at time t. In order to describe its future position, one needs to know its position at every time and the velocity, but it is not easy to measure the velocity for every time t. It can be measured by taking relative differences of the position of the spring, but this can enhance noise and makes the problem much more difficult to handle. However, there has been established a theory that has its roots in nonlinear dynamics that allows us to manage such cases and extract conclusions for difficult dynamical systems. *Delay coordinate embedding* is a very useful way to tackle this problem and can provide a very good approximation of the internal dynamics of the system from the measurement of a single variable. More specifically, this approach reconstructs the space of the dynamical system giving equivalent and sometimes identical topological properties with the original system. For example, if we have the full measurement of one of the state variables, delay coordinate embedding allows us to reconstruct the state space of the system as if we knew the values of the other state variables. This idea was introduced by Taken's [12] in 1981 but I will not go into the mathematical background of it.

4 FINANCIAL DATA

Before mentioning Echo State Networks and the concept of reservoir computing it is helpful to underline some key concepts of financial data and form the idea that will be applied in the chapters to follow.

4.1 Introduction

Financial markets form a very complex network of different factors that appear some dependencies between each other. That means that some changes of one factor affects somehow the behavior of the others. The evolution of a stock, and of the financial market as a whole, can be considered to be a nonlinear dynamical system that seems to have non-random chaotic dynamics [10]. Since the financial market can be affected by many qualitative and quantitative factors that are not fully known to the researcher it is of high importance to include factors that can help the model find the appropriate spatio-temporal patterns. So, a multivariate input signal is going to be fed into the network using four of the attributes that describe a stock. Apart from multivariate inputs that can boost the accuracy of the network it has to be mentioned here that data preprocessing is a key factor to achieving a capable model. As mentioned above, financial systems are dynamical systems that change over time and is not easy at all to predict the future position of the systems. Therefore, it is of high importance to try to include the best possible data that carry the appropriate information. That means that the format of the data needs to be carefully chosen as well as the dimensionality of the input signal can depict at a certain level the phase space of the system. Consequently, a researcher must apply certain methods to change the format of the data so as to drive the network to the targeted signal. At this point I should mention that the techniques to be explained will not be applied at every experiment and further details will be given at the results section.

4.2 Data Preprocessing

Stationarity is a concept that is found always when it comes to financial time series analysis. Assuming a random process X(t), it is said that X(t) is stationary if the mean, the variance and the autocorrelation structure do not change over time as shown in the Figure 4-1 below.

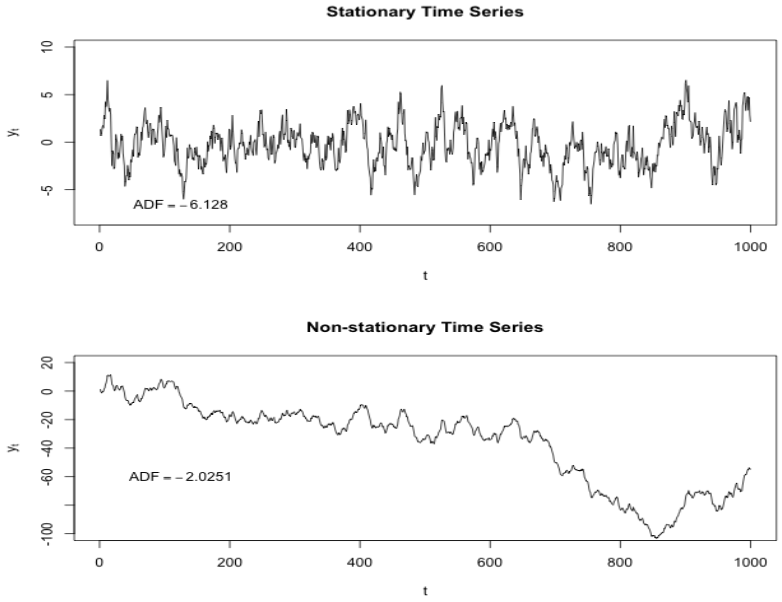

Figure 4-1. *Example of Stationary (Top) and Non-Stationary time series (Bottom)*

In a strict mathematical formula stationarity is interpreted like:

$$E\left[\left(X(t)-\mu(t)\right)^2\right]=\sigma=constant, \quad (4.1)$$

$$E[X(t)]=\mu=constant, \quad (4.2)$$

Further mathematical details about stationarity are not going to be discussed as it is not the purpose of this work. Stationarity can be achieved by many procedures some of which are going to be discussed. This process is also called detrending as it removes the trends and the cycles that are included in a time series. Differencing can be thought to be the simplest method of detrending even though sometimes it fails to make the mean constant, and it results in having the following time series based on an original time series Y(t):

$$Y(t)=[Y_2-Y_{1,}\ldots,Y_t-Y_{t-1}]$$

Another common practice in order to make a stochastic process stationary is relative differencing which results in having the time series:

$$Y(t) = [\frac{Y_2 - Y_1}{Y_1}, \dots, \frac{Y_t - Y_{t-1}}{Y_{t-1}}] \text{ , (4.3)}$$

Furthermore, another useful transformation can the application of the logarithmic function at forward differences or relative differences accordingly as follows:

$$Y(t) = \left[\log\left(\frac{Y_2}{Y_1}\right), \dots, \log\left(\frac{Y_t}{Y_{t-1}}\right)\right], \text{ (4.4)}$$

$$Y(t) = [\log(\frac{Y_2 - Y_1}{Y_1}), \dots, \log(\frac{Y_t - Y_{t-1}}{Y_{t-1}})], \text{ (4.5)}$$

If the above techniques are applied that would results in having really small values of the data and hence the W_{in} weights that connect the reservoir and the input signal should be scaled so as to exploit the nonlinear range of the sigmoid functions. Values close to zero when fed into a sigmoid function would lead to linear behaviour of the network and since financial data display nonlinear properties it is advisable to be used. When it comes to high dimensional data a good heuristic suggests scaling the input signal by a factor $\beta \sim \sqrt{d}$, where d is the dimension of the data.[10]

4.3 Metrics

For everything in life there should be a metric that plays the role of the judge when it comes to performance evaluation. In general, when it comes to financial forecasting, the main purpose is to predict correctly the trend of the quantity that is being under investigation. Consequently, the main metric that will evaluate the performance of the network is going to be the number of trends that were correctly guessed by the algorithm.

Trends define the direction of the stock on the next day. So, assuming that we have a test set of length *n* that produces *n-1* trends the indicator that shows how well the model performs is defined as, $correct\ trends = \frac{true\ guesses}{length\ of\ the\ test\ set - 1}$.

Another common metric that is going to be used is the Mean Square Error (MSE) that indicates numerically the performance of the model. MSE measures the average of the squared errors between the true outputs and the predicted ones. More specifically MSE is defined as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_{true} - Y_{pred}\right)^2, \text{ (4.6)}$$

According to MSE can be defined the Root Mean Square Error (RMSE) which can be calculated by taking the square root of MSE is a measure of how spread out the predicted values are from the regression line. Finally, the Minimum Absolute Error (MAE) which is also used in all the experiments and is the average absolute error between the predicted and the true outcomes of the algorithm. MAE is defined as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |Y_{true} - Y_{pred}|, \quad (4.7)$$

4.4 Preparation for the experiments

After the brief introduction about financial data, data pre-processing techniques and the metrics that will be needed for the experiments, a few words need to be said about the experiments. To begin with, the data that are used for the experiments can be found at yahoo.finance and are analysed using the MATLAB software package. More specifically, the experiments refer to the evolution of the stock price of the American Airlines starting from the 22nd of September 2009 up to 27th of September 2019. Furthermore, another dataset is used to evaluate the performance of the network and refers to the index S&P 500 that combines the stock prices of the 500 most influential companies worldwide. The training set of the American Airlines case consists of 2219 data points while those of S&P 500 consists of 2217. 199 data points are used for testing the algorithm for the latter case while the American Airlines dataset uses 204 data points. The first 100 data points of the datasets are used to initialize the reservoir state of the network. The goal is to predict the 'Close' price of the next day using the four attributes (Open, High, Low, Close) of the previous day and the Open value of the following day.

5 Reservoir Computing

At this point, after having mentioned neural networks it is time to introduce the reader to the concept of reservoir computing. In the next paragraphs of Chapter 5, reservoir computing and especially Echo State Network; one version of reservoir computing, is going to be analyzed in depth.

5.1 Introduction

Reservoir computing can be defined as another computational tool and shares many similarities with neural networks. The perception of reservoir computing reminds that of neural nets but has many differences with it and makes it a great tool especially for temporal and sequential data. The need that gave birth to this idea was the difficult and time-consuming part of the training of the Recurrent Neural Networks. So, about a decade ago Reservoir computing first appeared in two versions that have their roots in different fields; Echo State Networks that will be viewed further down and Liquid State machines. The philosophy of RC is quite like those of Neural nets. The input signal $u(t)$ is fed into what is called 'reservoir' at every timestep t using input weights W_{in} that connect the input layer with the rest network. The purpose of the reservoir and in fact what differentiates reservoir from a typical hidden layer is that, it maps the input signal into higher dimensions according to the size of the reservoir. Then with a simple readout algorithm like linear or ridge regression, the network performs the training part.

5.2 Echo State Network

The Echo State Network (ESN) is a variation of Reservoir Computing and was first introduced by Herbert Jaeger[13]. The concept of ESN is going to be given in detail in steps as simple as it can be so as to be easily understood by the reader. Echo state network is used as a supervised machine learning task for an input signal $u(t)$ and a true desired output Y_{true}. Firstly, as in neural networks, an input signal $\bar{u}(t)$, t $\in \mathbb{N}$, is fed into the network so as to initialize the network. Time t, as can be seen, is discrete and therefore each timestep corresponds to a unique situation. The input nodes of the network that receive the signal, are connected with the next part of the network called "reservoir" with weights that are randomly generated. So, there is a weight matrix W_{in} , that is randomly generated from a prior known distribution (e.g. Gaussian, uniform etc.) and connects the input layer with the *reservoir*. This part of randomness of the

weights differentiates echo state networks from other neural networks where the weights are learnt by the algorithm. This gives echo state networks a big advantage in terms of time and computational efficiency. Then, the reservoir nodes, like in the recurrent neural nets, are assigned a value based on an update equation that governs the dynamics of echo state networks. Each node follows the equation below:

$$x(t) = (1 - a) * x(t-1) + \tanh*\left(Win * [1; u(t)] + W * x(t-1)\right), t \geq 1 \quad , (5.1)$$

The above equation defines the state of each node of the reservoir for every timestep t. That means, that the input signal is mapped into a higher dimension according to the update equation and based on it reconstructs the phase space of the system. The choice of activation function is arbitrary and hence other activation functions can be chosen as well. Now that we have made a brief introduction on echo state networks, we shall examine the architecture of the network, the philosophy of the reservoir as well as explain the model's hyperparameters.

5.2.1 Reservoir

Echo State Network, as its name suggests, tries to preserve the memory of the already seen input data and tries to keep long-term dependencies, 'echoes' of past data. In fact, reservoir acts like a nonlinear, high-dimensional expansion of the input signal $u(t)$. Furthermore, it is an input-driven dynamical system in a way that the true output $Y_{true}(t)$ can be reached by a linear combination of it [14]. Practically, at each timestep t, the states of the reservoir nodes are updated according to the equation *($x(t) = (1 - a) * x(t-1) + \tanh*\left(Win * [1; u(t)] + W * x(t-1)\right), t \geq 1$, (5.1)* and are stored in a matrix whose columns refer to timesteps and rows to the values of the reservoir nodes. It is obvious from the formula, that $tanh$, or other sigmoid functions, play an important role as they squash the values between specific interval preserving that way the nonlinearities of the data and avoid having outliers that might mislead the network. As for the size of the reservoir, there is not a general rule that determines it. In general, the bigger the size, the better the results might be but that is not always the case as we will see later in the research. Obviously, every task demands different handling but in general a big reservoir might make it easier to find linear correlations between the input and the true output.

5.2.2 Weight matrices and Training

Weights in the echo state networks are randomly assigned and act like the weights of the neural nets. Input weight matrix W_{in} contains all the input weights that connect the nodes of the input signal with the reservoir nodes. They are not learnt by a supervised learning algorithm but, are assigned in advance and are drawn from a prior distribution. The weight matrix W_{out}, contains all the weights that connect the reservoir nodes with the desired output node. The different thing in the Echo state networks is though, that the only training part takes place in the end. More precisely, the Figure 5-1 below depicts the difference between these two concepts [14]. The output matrix W_{out}, is learnt once by the algorithm, by solving a linear equation based on the fact that the output is a linear combination of the reservoir states multiplied by a weight matrix. To be more mathematically correct the training part holds as follows. Just like in the neural networks, there is a cost function that measures the difference between the true output Y_{true} and the predicted output Y_{pred} by the reservoir. In this dissertation, Mean Absolute Error is used along with other metrics but for explanation reasons of the training process, let us assume that the cost function is the Mean Squared Error. Then, since reservoir is a linear combination of the output weights and the reservoir states, it holds that:

$$Y_{true} = Wout * X, \text{ (5.2)}$$

where W_{out} is the matrix of the output weights. Furthermore, it needs to be underlined here that $Y_{true} \in R^{N_y \, x \, T}$ and X $\in$ $R^{1 \, xN_u \, x \, N_x}$ is the concatenated matrix $[1; u(t); x]$ where x is the one column matrix that refers to the reservoir states for every timestep t of the training process. The purpose is to minimize the Mean Squared Error which is the difference between the predicted output and the desired one. System $Y_{true} = Wout * X$, (5.2) is an overdetermined system since T is much bigger than the product $(1xN_uxN_x)$ and there are many ways to solve such systems. The most common way is by using what is known as Tikhonov regularization which gives the following solution to system $Y_{true} = Wout * X$, (5.2)*:*

$$W_{out} = Y_{true} * X^T (X * X^T + \beta * I)^{-1}, \text{ (5.3)}$$

where β is the regularization coefficient and I the identity matrix.

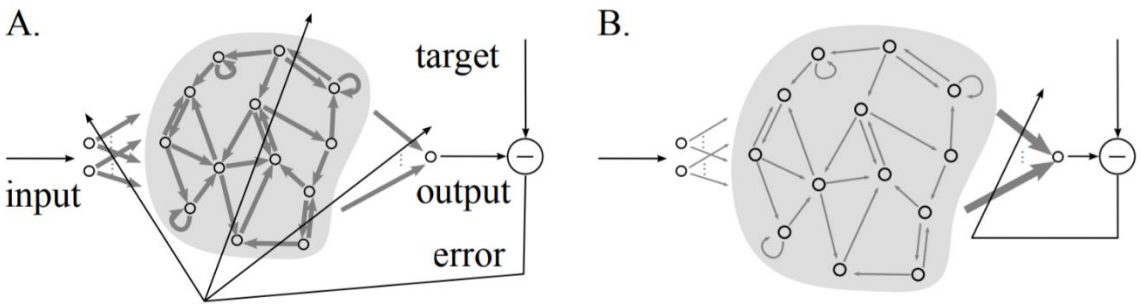

Figure 5-1. *A. The training part of a neural network, B. The training part of an ESN*

5.3 Leaking Rate

Leaking rate a, is a hyperparameter of the echo state network, taking place in the update equation of the reservoir nodes. In fact, if we consider the update equation $x(t) = (1 - a) * x(t-1) + \tanh * \big(Win * [1; u(t)] + W * x(t-1)\big), t \geq 1$, (5.1) as an Ordinary Differential Equation (ODE) for continuous time t, then it would be modelled as follows:

$$\dot{x} = -x + \tanh\,(Win[1; u(t)] + W * x),$$

The analogous expression for the discrete time t, can be achieved by Euler's discretization by taking:

$$\frac{\Delta x}{\Delta t} = \frac{x(n+1) - x(n)}{\Delta t}, \text{ (5.4)}$$

So, leaking rate can be perceived as the time interval between two timesteps, in the discrete case, in a row. It defines the speed of the dynamics of the reservoir nodes and hence, as its names suggests, affects how much of information leaks out of the reservoir affecting that way the short-term memory of the network.

5.3.1 Spectral radius

One of the weight matrices that has been previously mentioned, is the matrix denoted as W, which contains the weights that connect the reservoir nodes. This matrix is also randomly chosen, and the weights are not learnt by the algorithm. Spectral radius ρ, of matrix W is defined as its largest absolute eigenvalue and can be interpreted as the scaling of the matrix. In practical terms, spectral radius is one the most important parameters for echo state networks as it can affect along with the leaking rate the memory of the network. It decides how fast, the

effect of the input signal $u(t)$ for every timestep t, is washed out by the network and how stable the reservoir activations are [14]. At this level it should be mentioned that echo state networks are linked with an important property about spectral radius called the "echo state property". First researches showed that, for the network to be stable the spectral radius of matrix W should not be more than one ($\rho < 1$) which in simple words means that the network should not be affected by the history of long input sequences $u(t)$. But it was later argued and also established that the network can be also functional even if the echo state property is not satisfied. This can happen for non-zero inputs and can be explained by the use of an activation function that supports nonlinearities (e.g. tanh, sigmoid etc.) [14]. Intuitively it holds that spectral radius should be a big number when it comes to sequences that depend on long history. On the other hand, for sequences that depend on the recent past spectral radius should be a small number. The theoretical background of both global parameters and hyperparameters is necessary to establish the idea behind ESN but the values that these parameters can get vary from task to task.

5.3.2 Scaling

In order for the network to be optimized it is advisable to follow scale the weights of the input matrix. Since W_{in} is randomly chosen, it is common to use the normal or the uniform distribution. The first column of the matrix corresponds to a bias and therefore it is advisable to be scaled separately from the rest of the columns. Usually, when uniform distribution is chosen in a range [-α, α], the scaling factor is twice the range of the interval whereas in the normal distribution case one can use the standard deviation of the distribution as the scaling factor [14]. Furthermore, like in neural networks and since Reservoir Computing is part of machine learning it is also advisable to normalize the data before feeding them into the network as data are kept bounded and outliers are avoided.

6 Results

The purpose of this dissertation is to introduce an echo state network in order to forecast real financial data. More specifically, another approach is suggested as the idea is to use the Open, High, Low and Close attributes of a stock price at time *t* and also the Open attribute at time *t+1* so as to predict the Close price of the stock at time *t+1*. But before presenting the results and the architecture of the network it would be beneficial for the reader to understand how a simpler echo state network works for a macro-economic model proposed by Vosvrda et.al [15] which exhibits chaotic behaviour for certain selection of hyperparameters of the model.

6.1.1 Vosvrda

At this point it is now time to introduce Echo State Networks in practice and observe the behavior of the network both on real financial stock data but also data that derive from other sources. Such an example is an idealized Macro-economic model proposed by Vosvrda [15] which is modelled by a set of three first-order nonlinear equations. The data points that derive from the model will be used as an introductory case to ESN.

The Vosvrda system as it is mentioned above consists of three nonlinear equations which are solved using the MATLAB software package and the data are used to predict the next point given the previous. The three first-order nonlinear equations describe three different quantities as explained below:

$$\frac{dS(t)}{dt} = aY * (t) + p * S(t) * \big(k - Y(t) * Y(t)\big), \text{ (6.1)}$$

$$\frac{dY(t)}{dt} = u * (S(t) + F(t)), \text{ (6.2)}$$

$$\frac{dF(t)}{dt} = m * S(t) - r * Y(t), \text{ (6.3)}$$

where $S(t)$ are the household savings, $Y(t)$ is the GDP and $F(t)$ is the foreign capital inflow. The constants that appear in the equation are chosen based on the paper proposed by Vosvrda. The Vosvrda system of equations displays chaotic behavior, meaning that a small change in one of the parameters can lead to tremendously different results. More precisely, the $S(t)$ time-series that describes the household savings is tested on different

values of the parameter a and the Figure 6-1 is presented below as was suggested by Antoniades et.al [16].

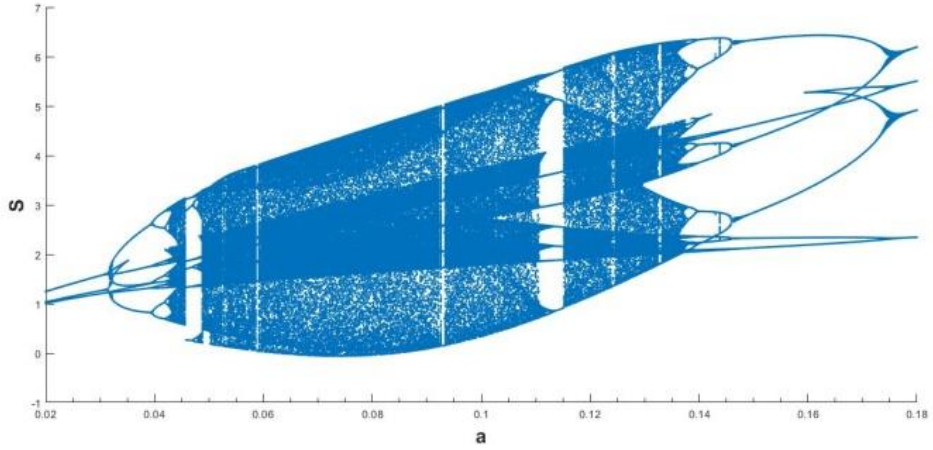

Figure 6-1. *Bifurcation diagram of $S(t)$ versus parameter a*

Bifurcation diagrams are used to show graphically the chaotic behavior of the examined equation. On the x-axis are the different values of the parameter of the equation and on the y-axis are the values of the equation. Basically, at the dense part of the graph it is showed at which values of the parameter the behavior of the equation is chaotic. When the graph is clearer and is a smooth line or a curve, the equation displays a periodical behavior and its period is the number of times an imaginary vertical line, at a specific value of a, inter-crosses with the graph.

6.1.2 Data Preprocessing

The set of data points that derive from the solution of the system is used to introduce the reader to the concept of Echo State Networks and how they work. The dataset consists of 6083 data points and the idea is to predict the next data point based on the previous one. The dataset is split into the initialization set which is set to 100 data points, the training set that is 4000 datapoints and the test set that consists of 500 points. The reservoir of the network is initially set to zero meaning that the value of every unit of the reservoir is zero. The input layer of the network receives at every timestep t a data point of the dataset which is mapped into the higher dimensional space using the W_{in} and the update equation. Every data point that is fed into the network and is mapped into the higher dimensional space affects the behavior of the reservoir which starts initially from the zero-state meaning that every unit is set to zero. Starting the

training immediately could lead to unstable solutions and hence the first one-hundred data points are left out so as to wash out the initial memory that the network gains from them. Many experiments were conducted in order to find the best hyperparameters of the network that would give the best approximation of the test set and are presented below.

W_{in}	$[-0.5,0.5], uniform\ distribution$
W	$[-0.5,0.5], uniform\ distribution$
$reservoir\ size$	150
$leaking\ rate$	0.1
$spectral\ radius$	1.25

Table 1. *Optimal hyperparameters for the Vosvrda systems*

For these hyperparameters the network performs extremely well as it can approximate the test set with a great accuracy. Specifically, the metrics that are used to judge the efficiency of the model are the successful trends and the Mean Squared Error. Since the test set consists of 500 data points this means that there are 499 changes from the i_{th} data point to the $(i+1)_{th}$.The Echo State Network guessed successfully 497 out of 499 times the direction of the next data point achieving a Mean Squared Error of 0.01. It can be also observed graphically that ESN approximates the test signal as it is the original one.

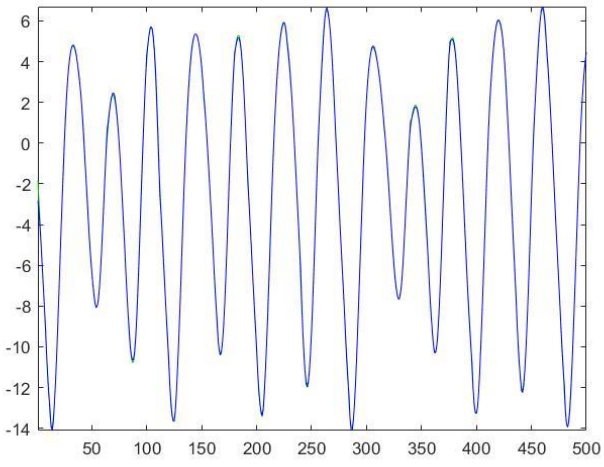

Figure 6-2. *Predicted values*

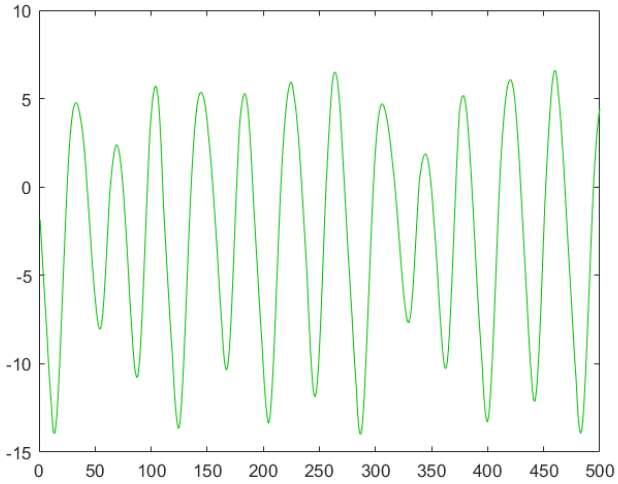

Figure 6-3. *Real values*

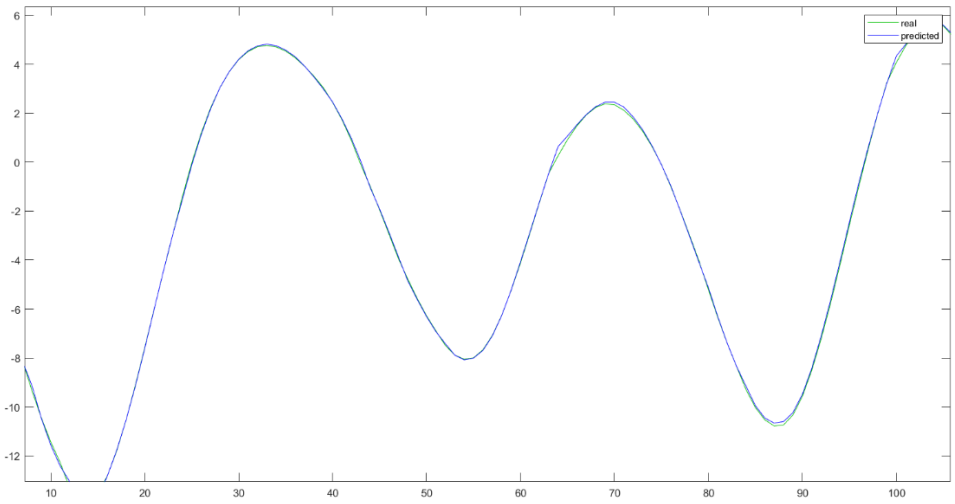

Figure 6-4. *Zoomed combined plot of real (green) and predicted values (blue)*

The reason that the Vosvrda systems is chosen to be tested using the Echo State Network is because it is a chaotic time-series that for given values exhibits smooth periodicity and can be used to introduce the concept of the network. For the ESN such time-series data can be approximated in detail and most importantly very fast. The training of the network lasts only a couple of seconds and is a great advantage against other computational tools. Now that a brief introduction is made and the theory of reservoir computing is established, it is time to deal with real financial data and see how the network performs on such cases.

6.2 American Airlines

After having introduced how an echo state networks behaves for simple free-noised data, the next step is to investigate and try forecast real financial data. The stock price of American Airlines is used as a pure stock but also the network will be tested for the index S&P 500. The data for American Airlines that were used for the research are for a specific time interval starting from 22/09/2009 to 27/09/2019. The approach of the network presented has its roots in nonlinear techniques and especially delay coordinate embeddings. We consider the evolution of the stock price as a dynamical system whose state variables are not known in advance. This means that in order to model and predict the system it takes a long investigation of the data to

capture its dynamics and patters. The only things that are available to the researcher are attributes that describe a stock: Open, High, Low, Close, Adjusted Close, Volume attributes. In my research the last two attributes were removed as they were considered irrelevant extra info to the model. This was because the Adjusted Close value was many times equal to the Close attribute and Volume was measured differently from the others and this affected the behaviour of the network. So, based on these four attributes that are available the idea is to exploit the reservoir of the echo state network and try to map them into the higher dimensional space so as to approach the number of the state variables. The lack of data due to the nature of financial data but also because of the unpredictability of the whole stock market led to the creation of the following network in an attempt of predicting the Close price of the stock of American Airlines.

In the model created for the American Airlines case 2517 data points were used. Specifically, the training was performed form the 101st till the 2317th data point using the four attributes mentioned above at time t and the Open value of the stock at time $t + 1$. The idea of leaving the first 100 data points out the training part comes from the fact that the echo state network starts from an initial state equal to zero. Hence, in order to stabilize its performance and also to wash out the initial memory of the network, the first 100 points are left out to serve this purpose. The idea of the reservoir, as it also mentioned previously, is to map the data into higher dimensions and then connect them with the output in a linear way.

This simply means that: $Y_{pred} = Wout * X$ and the idea is to minimize the Mean Squared Error (MSE) between the Y_{pred} and Y_{true}. This is an overdetermined system of linear equations because the number of rows of matrix X is much less than the number of the columns and there are plenty of options one can choose to solve such systems. In this work, I choose to implement ridge regression with Tikhonov regularization β [14] and have the final W_{out} matrix:

$$W_{out} = Y_{true}\, X^T (XX^T + \beta I)^{-1} \quad (6.4)$$

, where X^T is the transpose matrix of X, I is the identity matrix and β the Tikhonov regularization coefficient. Then, after computing the W_{out} matrix using the ridge regression method, the model was tested on 204 data points from which the last five correspond to the week from 23rd of September up to the 27th of the same month. It has to be mentioned here that the metrics used to judge the performance of the network are, the trend of the 'Close' value of the stock and the Mean Absolute Error (MAE).

6.2.1 Data Preparation

The data are fitted into the network after having been denoised using the built-in MATLAB function and normalized in a specific range. ESN has many hyperparameters that can be tuned according to examined task in order to find the optimal network. In this work the experiments of certain hyperparameters are going to be presented and discussed.

Before going through the results, I should underline the importance of denoising the data just by taking a close look at the following figures. Noise in financial data exists and can be interpreted by the lack of data, external factors that affect (in)directly the examined quantity and above all by uncertainty. For the purposes of this work, the simplest denoising function of *MATLAB* is used in order remove noise from the data. Figure 6-5, shows the noisy original data whereas Figure 6-5, shows the denoised one. It can be spotted that the denoised version of the data seems to be smoother and does not perform many steep ups and downs. So, visually it can be easily understood why this step is helpful in order to conduct the research. Further down, it will also be understood by comparing the MAE scores for both versions of the data. The concept of reservoir computing and especially the application of Echo State Network seems to be extremely promising and the results are going to be presented in the following lines.

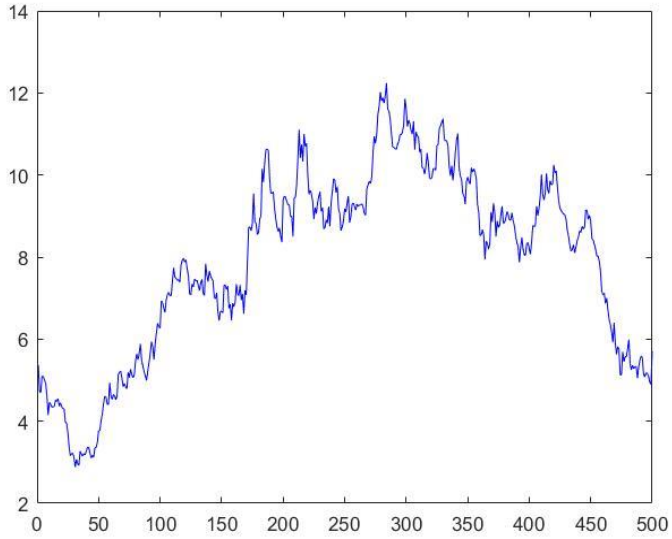

Figure 6-5. *Noisy original data*

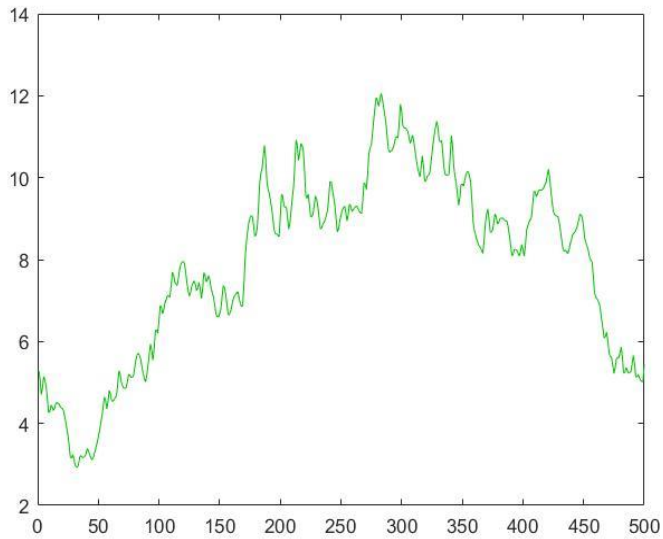

Figure 6-6. *Denoised Data*

6.2.2 Model

The attempt is to predict the time-series that are created by the 'Close' value of this attribute of the stock. In other words, many input variables are used in order to predict one-by-one the data points of a univariate time-series. So, at the beginning the experiments were conducted by taking as an input signal four different attributes of the stock that took place at day t and also the Open attribute of day $t + 1$. In fact, the algorithm is capable of producing predictions for the Close price of the stock straight after the stock market opens every morning. It is a common fact that the stock market is affected by the recent past and hence the algorithm was extended in order to take advantage of the four attributes of the stock of previous days.

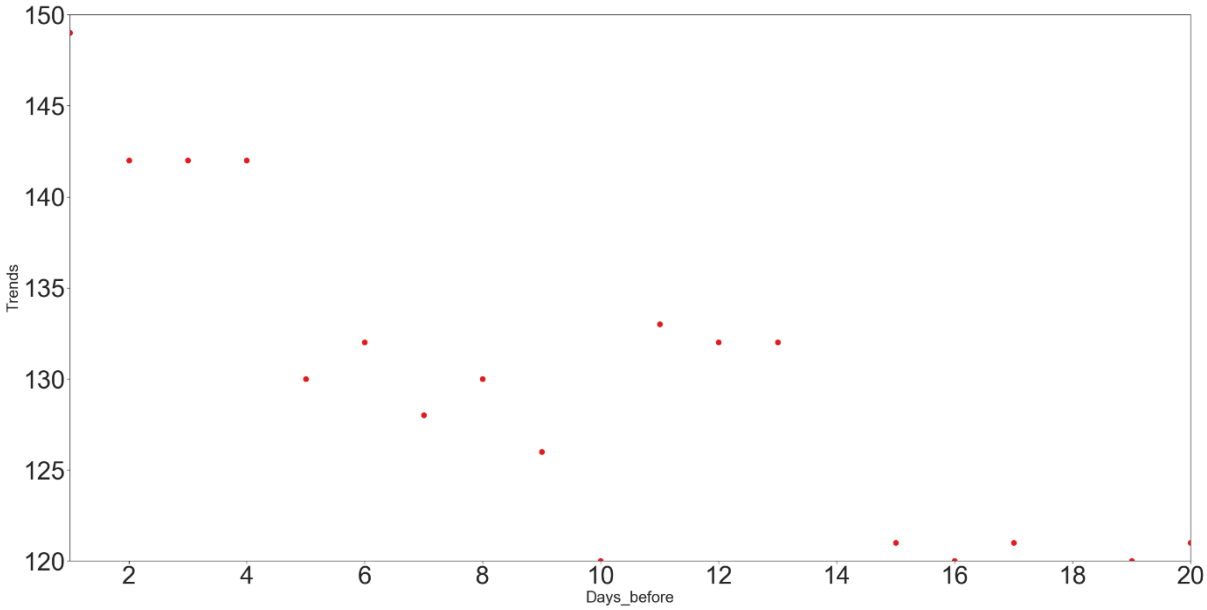

Figure 6-7. *Plot of successful trends (y-axis) and the days before that are used (x-axis)*

6.2.3 Results

The plots below depict that the best results in terms of successful trends that are obtained when one and two previous days are fed into the network plus the value of the Open attribute for the day of interest.

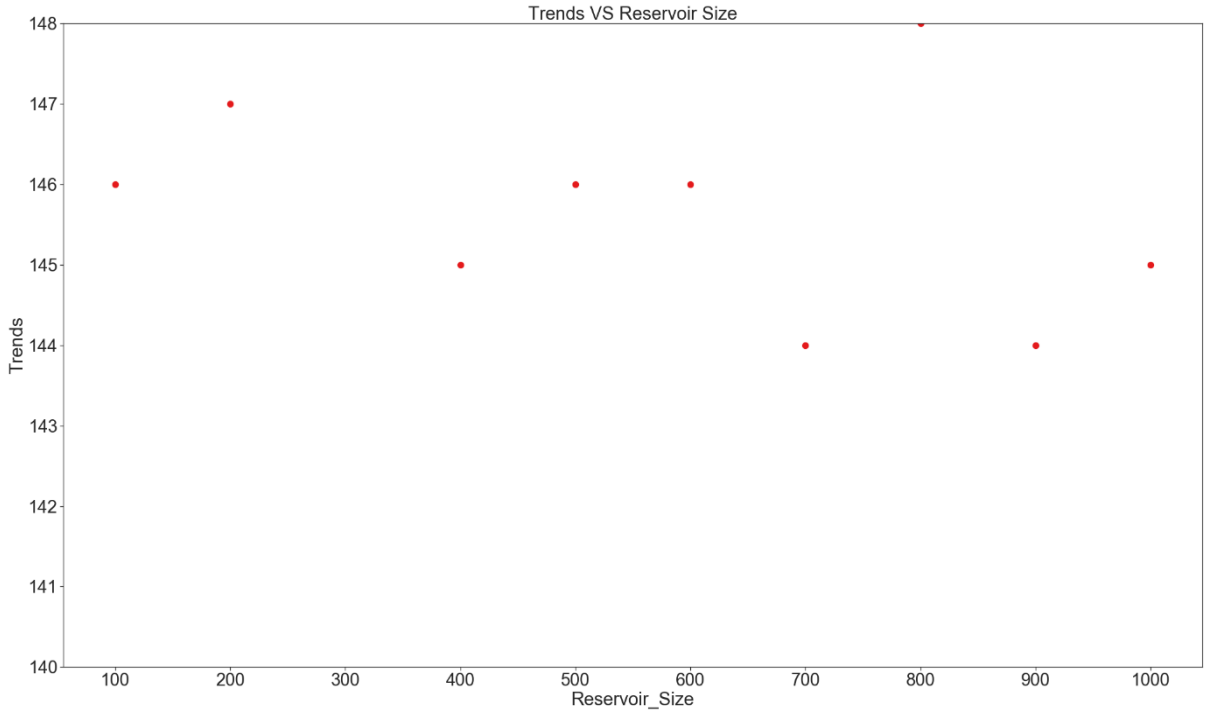

Figure 6-8. *Plot between reservoir size and trends for rolling window equal to one*

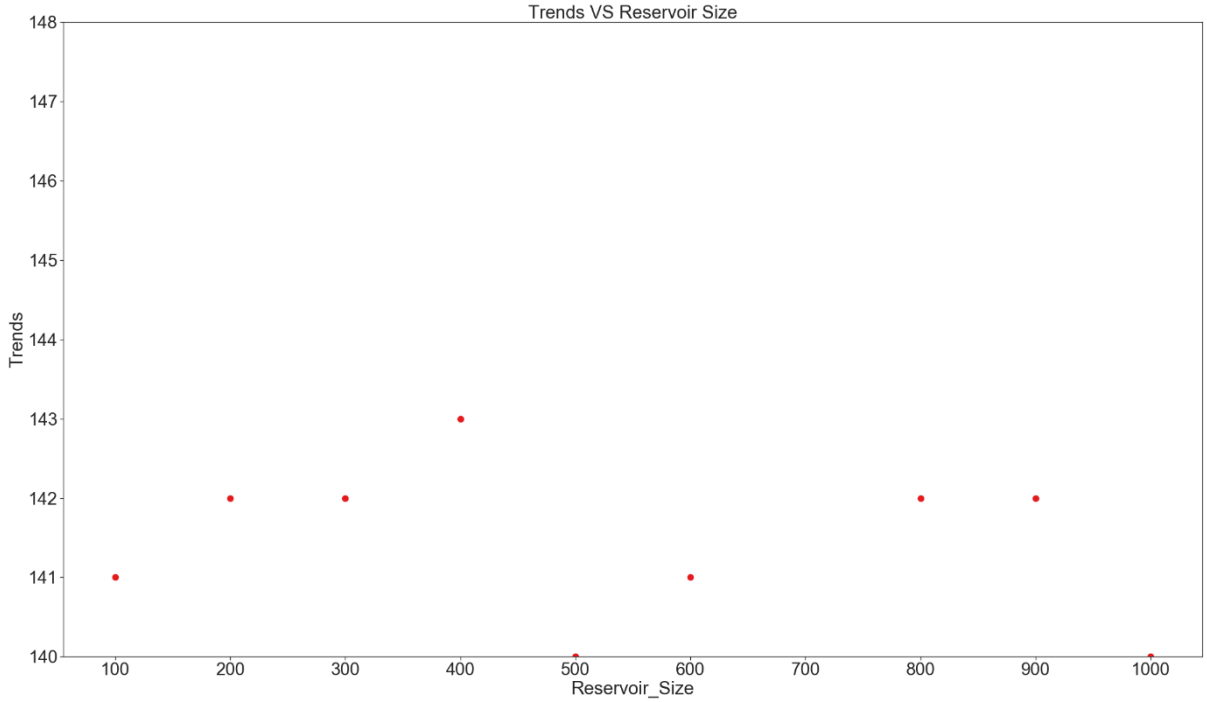

Figure 6-9. *Plot between reservoir size and trends for rolling window equal to two*

It becomes obvious that the best results are achieved when the algorithm accepts an input signal composed of four attributes at time t (Open, High, Low, Close) plus the Open attribute at time $t + 1$ for reservoir size equal 300.

It should be underlined that the pre-processing technique of denoising the data is helpful because it transforms the data to a smoother version and makes it easier for the network to capture its dynamics. Below there is another plot that indicates the importance of denoising the data. Numerically the difference between the two processes also show the importance of denoising. With denoising the algorithm predicts the trend 149 out of 203 times whereas in the non-denoising case the algorithm succeeds only 121 times.

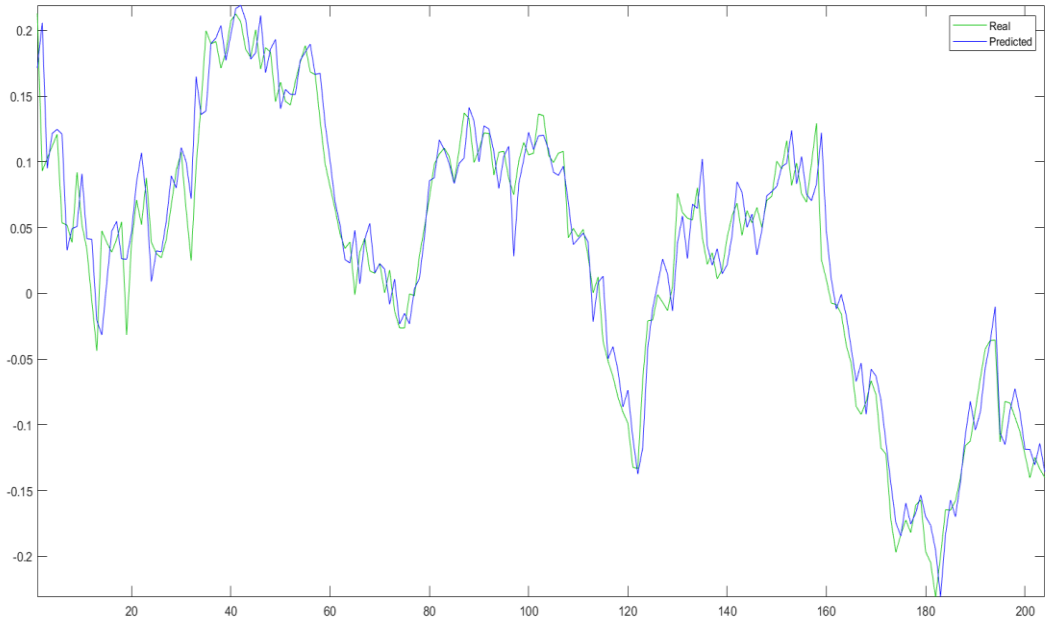

Figure 6-10. *Plot of real and predicted data without denoising*

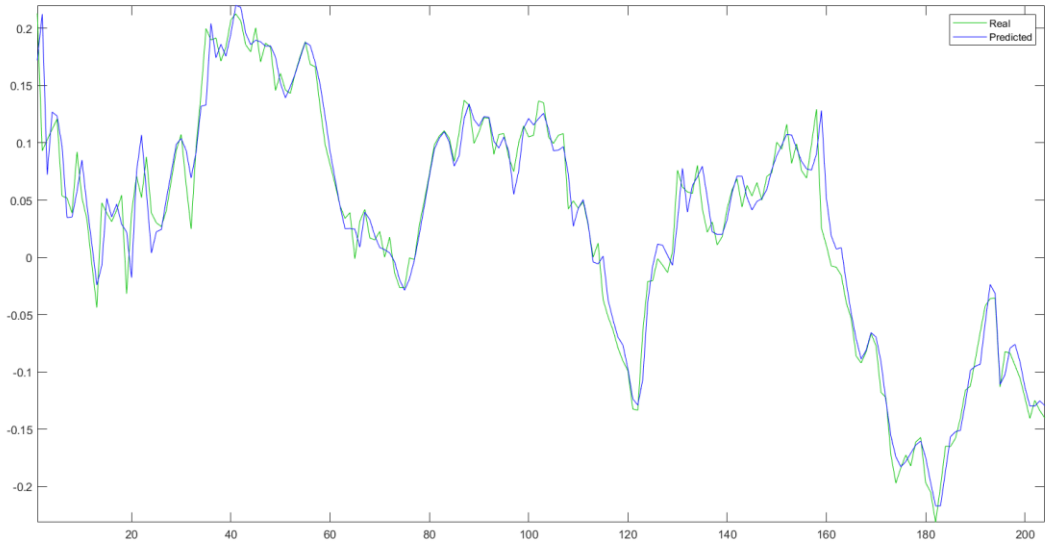

Figure 6-11. *Plot of real and predicted data with denoising*

Another key concept in achieving the best approximation of the output signal is the initialization part before the training process. The absence of the initialization part could lead to unstable behaviour of the network and this is depicted in the figure below. There can be observed a complete shift in the data meaning that the algorithm uses as a prediction the exact previous data point. This is simply because the network starts to learn the patterns of a complete random process immediately without letting the reservoir to move away from the initial zero-state. For that reason, the first 100 data points are left out and the best approximation is achieved.

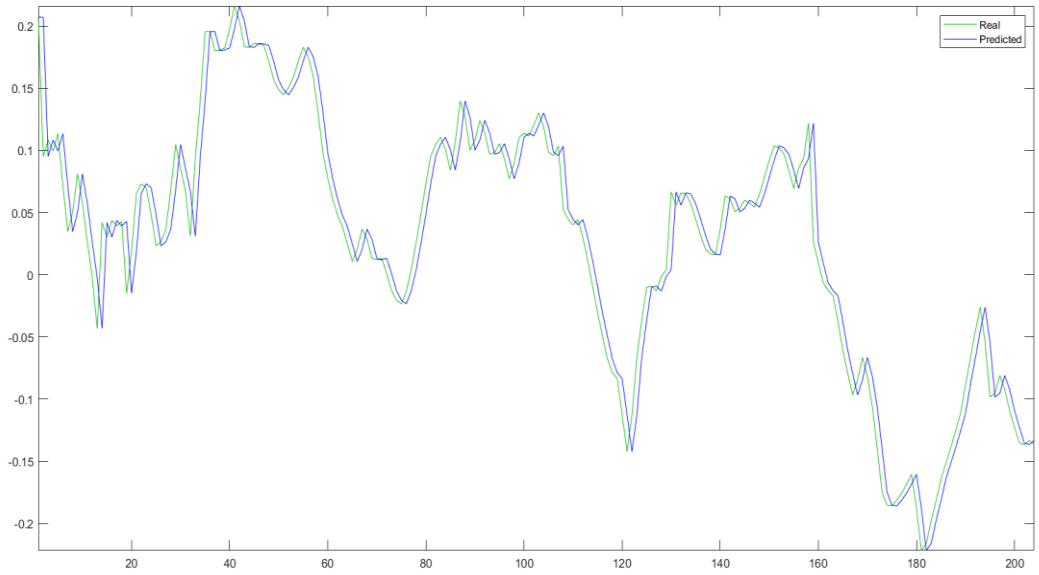

Figure 6-12. *Plot of the predicted output signal with no initialization part*

$Reservoir\ Size$	300
$Spectral\ Radius$	0.7
$Leaking\ Rate$	0.5
Win	$[-1,1], uniform\ distribution$
W	$[-1,1], uniform\ distribution$
$Activation\ Function$	$tanh$
MAE	0.0146
$Trends$	$149/203 \approx 73\%$

Table 2. Optimal values of hyperparameters for the American Airlines dataset

It is a common fact that the tuning of the hyperparameters of the model can be an extensive process as small changes in a quantity could lead to completely different results. For example, leaking rate and spectral radius are responsible for the memory of the network and can perform differently if tuned separately or together. The reservoir size is another important parameter of the Echo State Network as it defines the new higher dimensional space of the input signal and the training time. It has to be underlined here that in the pre-processing techniques the data retain their format and are converted into a stationary process like in many other time-series analyses techniques. However, the approximation seems to be really well, and the network can guess approximately the trends 73 per cent of the times. As Konrad states [10], the value of a stock cannot vary from day-to-day more than 3% which leads to comparing the results of the network to a complete random guess. Assuming a random process that follows a Gaussian distribution, we draw for every time t a sample from the Gaussian distribution with mean μ the real value of the stock and standard deviation equal to $0.01 * \mu$. The standard deviation comes from the fact that the 99.74% of the observations belong to the interval $[-3\sigma, 3\sigma], \sigma\ is\ the\ standard\ deviation.$ Solving either of the equations $\mu + 0.3 * \mu = 3\sigma$ or $\mu - 0.3\mu = -3\sigma$ we conclude that $\sigma = 0.01 * \mu$.

Compared to a complete random guess the echo state network outperforms as it can guess 42 more correct trends. More precisely, in the table below the results between the two processes are presented.

Process	*MAE*	*Trends*
ESN	*0.013*	*149*
Random Guess	*0.017*	*107*

Table 3. *Comparison between random guesses and echo state network*

6.3 S&P 500

6.3.1 Introduction

In the second part of the results the approach of predicting the 'Close' value of a stock differs from the first part. This time, an index is under research, but the goal is still the same; to predict the 'Close' price of the index. An index reflects a measurement of the value of the stock market based on average values of different stocks and hence can be affected easily by expensive stocks. Indices are used by the experts to evaluate the situation of global stock market and make judgements and reports. It is not possible to invest directly on an index, but it is possible to invest on a quantity that is indirectly connected with an index. In the results below, the experiments are conducted using the S&P 500 index which is characterized by many experts as the most representative metric of the stock market. The S&P 500 index, as its name suggests, combines information based on the stocks of the five hundred most influential companies.

Process	*MAE*	*Trends*
ESN	*0.013*	*167*
Random Guess	*0.017*	*133*

Table 4. *Comparison between random guesses and echo state network for S&P 500*

6.3.2 Model

In the American Airlines case, the goal is to use the certain four attributes of a stock that took place at day t plus its Open value at day $t + 1$ and predict the Close value at the same day. The situation in the S&P 500 case remains the same, but an attempt of providing certain measurements of the index is going to be made. The time-series is not converted to a stationary one using methods proposed in Chapter 4 but retains its form and dynamics. The pre-processing technique is applied before feeding the input signal into the network but this time the data are not scaled in a specific interval $[-a, a]$. In general, reservoir computing, since it needs sigmoid functions for the update equation of the reservoir nodes, demands the data to be bounded so as to avoid outliers and not lead to unstable solutions. However, S&P 500 was used to test the performance of the model after manipulating the data only at a denoising level, meaning that no further pre-processing technique was applied.

The idea of keeping an amount of initial data points out of the training process is applied also here as it helps the network to retain the information of the initial data points and forget its initial zero state. More specifically, reservoir can exhibit chaotic behaviours and hence depends heavily on initial conditions. Hence, its initial state might lead to different outcomes and as a result the first one hundred data points are left out of the training process.

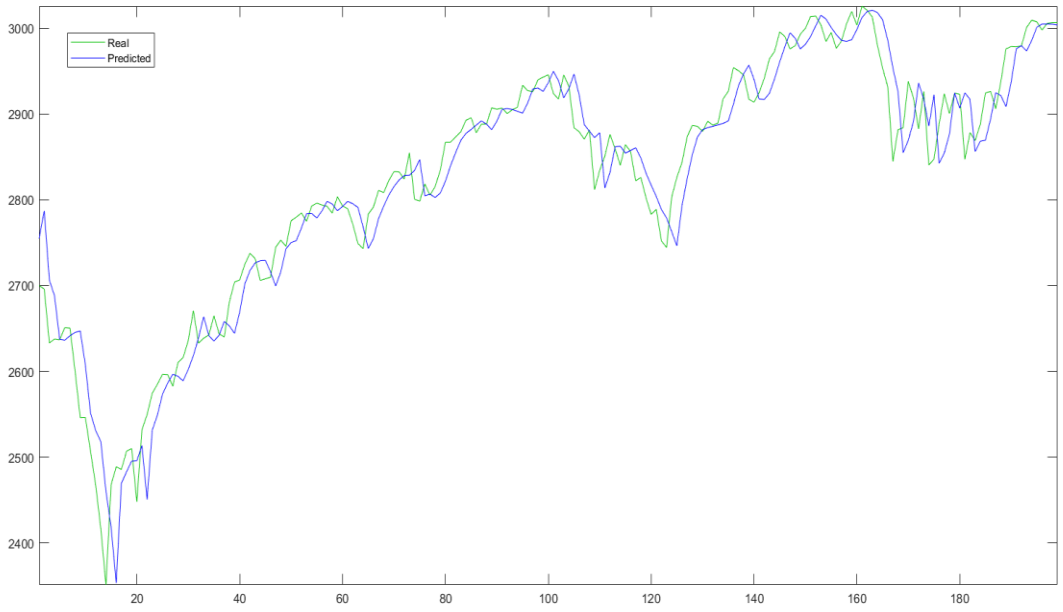

Figure 6-13. *Predictions (blue) with no initialization*

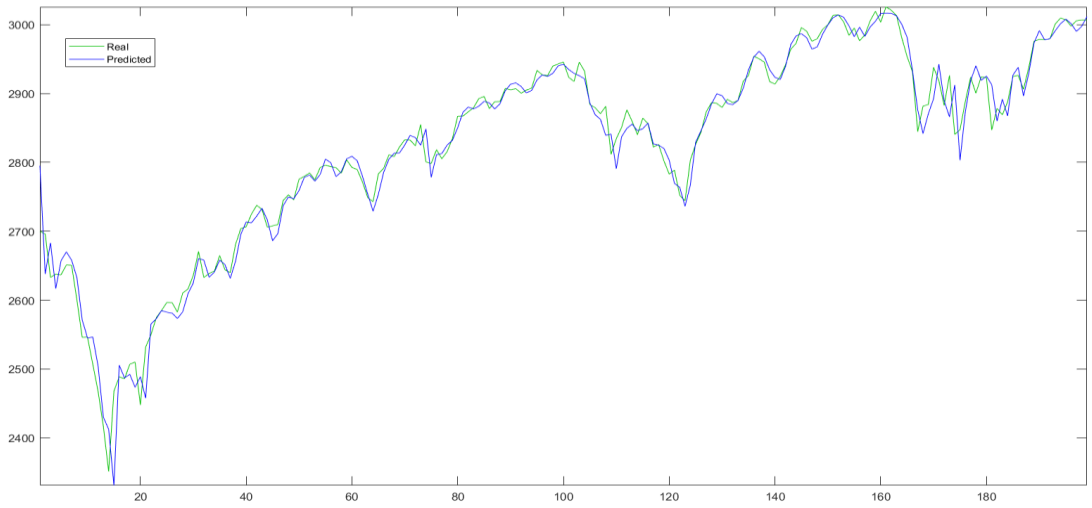

Figure 6-14. *Predictions (blue) with initialization*

6.3.3 Results

It becomes obvious that for no initialization there exists a complete shift in the data meaning that the network randomly proposes as a prediction for timestep $t+1$ the exact data point at time t. Of course, this cannot be a proposal of a successful algorithm and so the initialization limit is set to one hundred. In practical terms, the training set consists of 2219 data points and the test set of 199 data points. The data used for the experiments can be found online at *yahoo.finance* and correspond to a specific time interval starting from 17th of September 2009 until 19th of September 2019. After introducing the plan of attack for the examined task, the results of the network are going to be presented in terms of Mean Absolut Error (MAE) and successful trends. The key part of reservoir computing in this work is that the two time-series that are examined are not transformed into stationary ones. The pure raw data carry much noise, as it was mentioned above, because of their financial nature and the uncertainty of the global financial and geo-political situation. For that reason, the denoising of the data that will be used for training, is compulsory and key part of the algorithm's potential. Another point of interest in the research of the best approximation algorithm is the use of the activation functions in the update equation of the reservoir nodes. In the table below the results of using two different activation functions, $tanh = \frac{e^{2x}-1}{e^{2x}+1}$ and $sigmoid = \frac{e^{x}}{e^{x}+1}$, are presented keeping all the other hyperparameters of the model constant and the data are scaled in the $[-1,1]$ interval.

Activation function	*MAE*	*Trends*
$tanh$	0.0158	144/198
$sigmoid$	0.0146	145/198

Table 5. Comparison results between different activation functions

It can be seen that, there is not much different between the two activation functions since they perform equivalently well in terms of both MAE and successful trends. This can be explained by the interval that the data are mapped into which is symmetrical around zero and helps the two functions capture well enough the nonlinearities of the data. The quest is though to present a model that will be able to both predict the exact value of the index as well as suggest the direction the index will move towards to. For that reason, in the next experiments the presentation of such a model is going to be made along with optimization tasks that took place till the final choice of the proposed model.

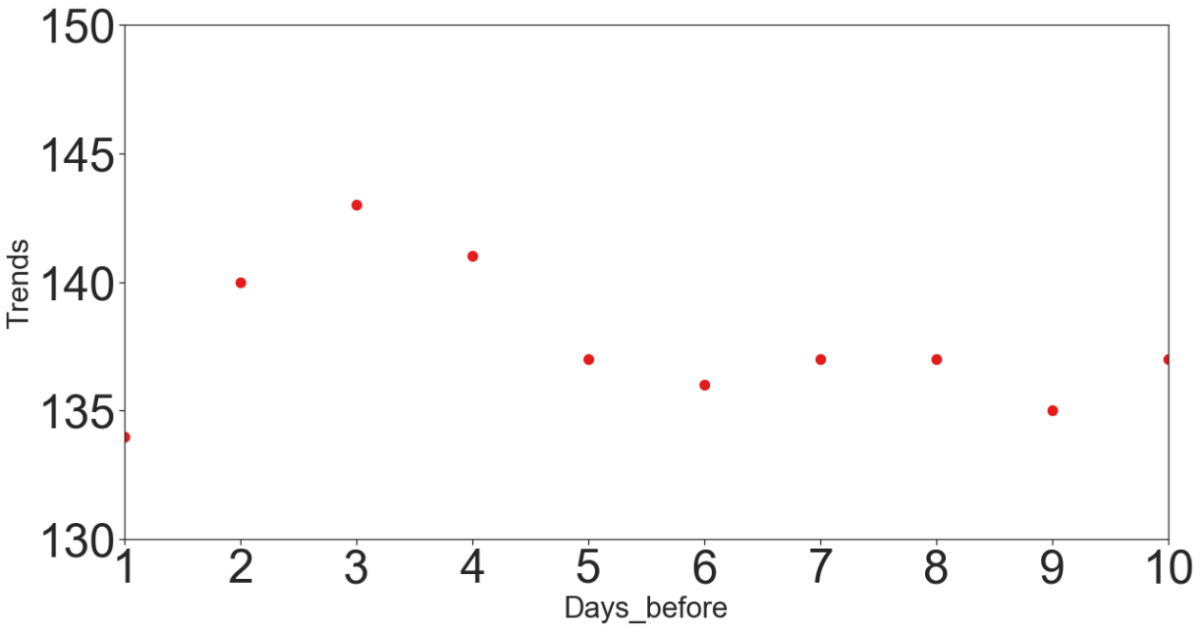

Figure 6-15 . *Days before that are used for prediction (x-axis) versus successful trends (y-axis)*

One of the key concepts when dealing with financial data, is the assumption that the behavior of the examined quantity is affected by its recent past. Along with that concept, the existence of leaking rate and spectral radius that were explained in Chapter 2, the network is fed consecutively with past information about the four attributes (Open, High, Low, Close). That means that for every experiment that it was conducted, the values of these four attributes of days $t-i, i \in N^*$, are fed into the algorithm. The results in terms of successful trends are presented below. So, in the proposed model uses the information of the past three days is used to produce the final results. Along with that valuable technique that leads to better results, the algorithm was also tested.

Since the data are not scaled, the goal is to find the best approximation between the true values and the predicted ones. So, the experiment is to find the scaling constant that refers constant the better the approximation is in terms of Mean Absolute Error. In the table below, the best optimization results are proposed along with the plot between the predicted and the original signal of the Close attribute of the index S&P 500.

Reservoir Size	300
Leaking Rate	0.5
Days before used for the prediction	3
W_{in}	[-1,1], uniform distribution
W	[-1,1], uniform distribution
Spectral Radius	0.7
Activation function	tanh
Scaling constant	10

Table 6. Optimal hyperparameters for the S&P 500 dataset

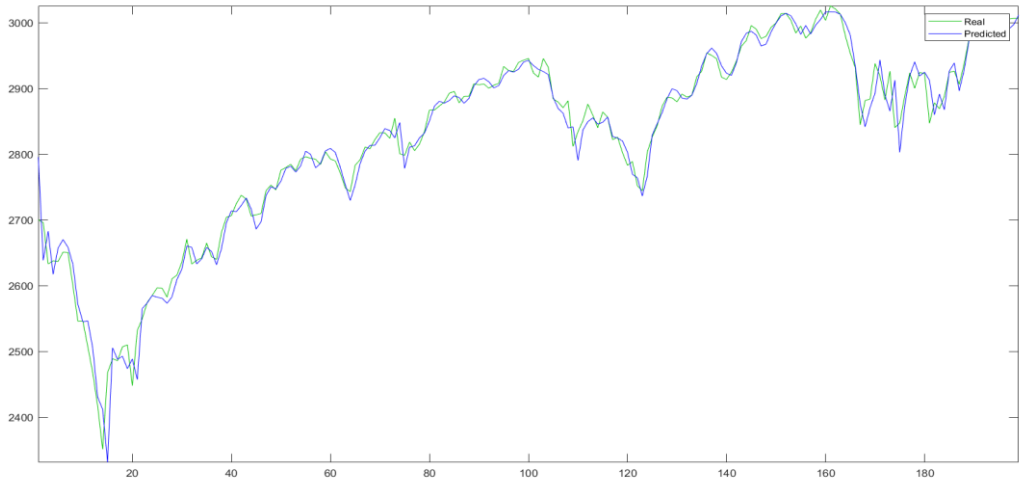

Figure 6-16 . *Plot of the predicted and the original data of S&P 500*

For these specific values, the proposed model achieved a score in terms of MAE of 14.7061 and succeeded in correctly predicting the trend in 143 out of 198 times.

At this stage, I should mention that in order to judge the performance of the model, it is compared to a random guess from a gaussian distribution as it was performed in the American Airlines case. The comparison between the two, suggests that the echo state network succeeds in predicting the trends more times than a simple random guess from the normal distribution. More specifically, completely random true guesses of the trends are 98 out of 198 which is much less than those of the echo state network. The philosophy remains the same and is based on the fact that every next value of the index cannot vary more than 3 per-cent. Furthermore, another comparison that shows the potential of the algorithm is performed based on the results of the experiments conducted by the fellow colleague Spyridon Georgopoulos. His results showed that the equivalent dataset performs worse using Recurrent Neural Networks (RNN) in the scaled version of the data in the interval [0,1]. The comparison is accumulated in the following table.

Algorithm	Trends
ESN	143/198
RNN	117/198

Table 7. Comparison between ESN and RNN

As it is mentioned above, the model for the S&P 500 case was built without rescaling data, allowing the algorithm to make exact prediction of the values of the 'Close' price. Below it can be seen the true and the predicted values that correspond to the last five values of the test set, where the algorithm successfully identifies 3 out of 4 trends.

Day	True	Predicted
13/09/2019	3007.4	3008.3
16/09/2019	2997.9	3001.9
17/09/2019	3005.7	2990.4
18/09/2019	3006.7	2997.4
19/09/2019	3006.8	3011.7

Table 8. True and predicted values of the Close attribute of S&P 500

It becomes an undisputable fact that the echo state network performs really well on financial data and looks promising for further analysis in order to approximate better the original data.

7 Conclusions

To sum up the goal of this master thesis is to introduce the reader to the concept of reservoir computing and more specifically to Echo State Networks. The idea of reservoir computing comes from the field of nonlinear dynamics and chaos theory and hence there has been a reference to the fundamental theory of these fields. Financial data, due to their chaotic behavior and uncertainty, is the center of my research with the goal being to predict the Close price of the stock of American Airlines and the S&P 500 index. The Vosvrda chaotic macro-economic model showed that it can approximated extremely well by the Echo State Network as it successfully guessed 497 times out of 499 the trend of the signal. The results were confirmed to be good as well as far as the American Airlines' stock is concerned. The model was fed with the values of four main attributes of the stock (Open, High, Low, Close) at day t and the Open price at day t+1, and the goal was to predict the Close price at day t+1. The results showed that the algorithm correctly identified the trend of the stock 149 times out of the 203 total trends. The idea of using past information about the data showed that when the rolling window is equal to 1 the best results are obtained in terms of successful trends. Then, the network was also tested for the S&P 500 case, which is an index that describes the situation of the stock market. The results were also encouraging for the potential of the algorithm, as it successfully guessed the trend of the index 143 times out the 198. It has to be mentioned that in the latter case, the data were just denoised and not rescaled in a specific interval. Furthermore, in both situations the data kept their dynamics as they were not transformed in stationary time-series using specific transformations. This allowed the network to suggest the exact predicted values of the S&P 500 giving that way a compact prediction for the index.

8 Future Work

Reservoir computing seems to be promising and can perform well with financial data. However, the research can be extended in order to improve the performance of the network in terms of trends. One thing that could be examined is the behaviour of the network on stationary data that would make the time-series smoother and less noisy. Then, the application of rolling window, in a sense of using past data, could be also applied. Furthermore, it would seem interesting to train the data so as to produce predictions for more upcoming days and not only day-to-day predictions. Extending the horizon of predictions could help decision making and trading strategy as it would monitor the total returns better. Finally, another thing that could also help the network perform better could be the denoising of the data using appropriate software packages like TISEAN that are built for nonlinear dynamics to handle such situations.

9 References

[1] Wikipedia, “https://en.wikipedia.org/wiki/Time_series.”

[2] NIST, “https://www.itl.nist.gov/div898/handbook/pmc/section4/pmc4.htm.”

[3] “http://www.earth-policy.org/indicators/C51,” *earth policy*.

[4] Wikipedia, “https://en.wikipedia.org/wiki/Recurrent_neural_network.”

[5] M. Lukoševičius and H. Jaeger, “Reservoir computing approaches to recurrent neural network training,” *Comput. Sci. Rev.*, vol. 3, no. 3, pp. 127–149, 2009.

[6] E. Bradley, “Intelligent Data Analysis: an Introduction. Chapter 5 - Time Series Analysis,” *Book*, pp. 1–25, 2009.

[7] Scholarpedia, “http://www.scholarpedia.org/article/Mackey-Glass_equation.”

[8] J. Su, “Reservoir Computing in Forecasting Financial Markets,” 2015.

[9] Y. Wang, “Applications of Recurrent Neural Network on Financial Time Series,” pp. 2016–2017, 2017.

[10] K. Stanek, “Reservoir computing in financial forecasting with committee methods,” 2011.

[11] S. Viera, “https://www.researchgate.net/profile/Sandra_Vieira5.”

[12] Wikipedia, “https://en.wikipedia.org/wiki/Takens%27s_theorem.”

[13] E. S. Network, “http://www.scholarpedia.org/article/Echo_state_network.”

[14] M. Lukoševičius, “A practical guide to applying echo state networks,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7700 LECTU, pp. 659–686, 2012.

[15] Vosvrda, “Vosvrda, M. Bifurcation Routes and Economic Stability, Bulletin of the Czech Econometric Society 14, 43-60 (2001).”

[16] Antoniades. et.al, “Applying Complex Network Time-series Analysis to a Macro-economic Dynamical Model.”

Tables

Figures